

CLAIMS

1. A method for testing and verifying a requirements specification of a system, comprising:
describing said requirements specification in a REQUIREMENTS ENGINEERING LANGUAGE (REL);
simulating an execution of a scenario of said REL; and
identifying logical faults in said requirements specification based on said simulating.
2. The method of claim 1, wherein said describing further comprises separating said requirements specification into at least one entity to express a characteristic of said requirements specification.
3. The method of claim 2, wherein said separating said requirements specification into said at least one entity further comprises defining at least one type of a variable in said requirements specification.
4. The method of claim 3, wherein said separating said requirements specification into said at least one entity further comprises defining at least one entity to contain said at least one type of variable.
5. The method of claim 4, wherein said step of defining further comprises expressing at least of one constraint on said at least one type entity.
6. The method of claim 2, wherein said separating said requirements specification into said at least one entity further comprises defining a possible manipulation of said at least one entity during said simulating.
7. The method of claim 6, wherein said possible manipulation is defined by an input variable.

8. The method of claim 6, wherein said possible manipulation is defined by at least one condition that is required to exist for said possible manipulation to be done.
9. The method of claim 6, wherein said manipulation is done in parallel during said simulating.
10. The method of claim 6, wherein said manipulation is done sequentially during said simulating.
11. The method of claim 2, wherein said separating said requirements specification into said at least one entity further comprises defining an entity to describe said system.
12. The method of claim 2, wherein said separating said requirements specification into said at least one entity further comprises defining at least one temporal restriction imposed on said system.
13. The method of claim 12, wherein said defining at least one temporal restriction imposed on said system further comprises describing a behavior of said system over at least one cycle of said simulating.
14. The method of claim 13, wherein said defining at least one temporal restriction imposed on said system further comprises defining a possible event that may effect said system.
15. The method of claim 2, wherein said separating said requirements specification into said at least one entity further comprises defining a scenario to be simulated during said simulating.
16. The method of claim 1, wherein said simulating further comprises checking said requirements specification to be logical.

-
17. The method of claim 16, wherein said checking further comprises checking for no conflicting requirements.
 18. The method of claim 16, wherein said checking further comprises checking for no more than one possible outcome for an operation to be simulated or for a value of said at least one entity.
 19. The method of claim 16, wherein said checking further comprises checking for an outcome for said operation to be simulated or for said value of said at least one entity.
 20. The method of claim 16, wherein said checking further comprises checking at least one restriction defined by a user of said system.
 21. A system for testing and verifying of requirements specification of a system, comprising:
 - a modeling and testing component to build a model of said requirements specification; and
 - a dynamic testing component to test said requirements specification by execution of at least one simulation cycle.
 22. The system of claim 21 further comprising a main repository component to store said model of said requirements specification.
 23. The system of claim 22 wherein said main repository component is auxiliary to said system.
 24. The system of claim 21, wherein said modeling and testing component performs at least one static test to said model of said requirements specification.

-
25. The system of claim 21 wherein said modeling and testing component translates a high-level specification language to said requirements engineering language.
26. The system of claim 21 wherein said modeling and testing component receives said requirements engineering language.
27. The system of claim 21, wherein said modeling and testing component further comprises:
a moderator component to translate said high-level specification to said requirements engineering language;
an object builder component to build a unique representation of said specification requirements; and
a checker component to check characteristics of said unique representation.
28. The system of claim 22 wherein said main repository further comprises a rule repository component to store syntax rules of said high-level specification language.
29. The system of claim 22 wherein said main repository further comprises a specification repository component to store said specification requirements after said checker component checks it.
30. The system of claim 21, wherein said dynamic testing component further comprises:
a requirements tests manager to define, simulate, and analyze a scenario;
a simulator manager to coordinate the simulation sequence of said scenario;
a dynamic verification manager to activate said checkers during said simulation sequence; and

a simulation and verification manager to control said dynamic testing component.

31. The system of claim 30, wherein said simulation and verification manager is further adapted to load at least one entity for simulation from said specification repository component.
32. The system of claim 31, wherein said at least one entity can be simulated substantially immediately.
33. The system of claim 30, wherein said simulation and verification manager is further adapted to analyze said scenario.
34. The system of claim 30, wherein said simulation and verification manager is further adapted to initiate said simulator manager and load said checkers for said dynamic verification manager.
35. The system of claim 30, wherein said simulator manager further comprises:
 - an evaluation component to provide an infrastructure required to execute said scenario; and
 - an evaluation context component to control an execution of at least one evaluator of said scenario.
36. The system of claim 35, wherein said at least one evaluator is executed in parallel to at least a second evaluator.
37. The system of claim 35, wherein said evaluator is evaluating an entity written in said requirements engineering language.
38. The system of claim 35, further comprises an outcome repository to store the outcome of said execution of said at least one evaluator of said scenario.

39. The system of claim 30, wherein said dynamic verification manager is further adapted to receive values of said at least one entity during simulation.
40. The system of claim 30, wherein said dynamic verification manager is further adapted to check if said values are true values for said at least one entity within said simulation.
41. The system of claim 30, wherein said dynamic verification manager is further adapted to detect an error of said execution by said at least one evaluator of said scenario.
42. The system of claim 27, wherein said object builder further comprises:
a temporal entity object builder to receive a temporal logic entity from said modeling and testing component,
wherein said temporal entity object builder is adapted to divide said temporal logic entity to at least one component said temporal logic entity is built from; and
wherein said at least one component is sent to a builder within said object builder.
43. A modeling and testing apparatus comprising:
a moderator component to translate a high-level specification requirements to a REQUIREMENTS ENGINEERING LANGUAGE (REL);
an object builder component to build a unique representation of said specification requirements; and
a checker component to check characteristics of said unique representation.
44. The modeling and testing apparatus of claim 43, wherein said object builder further comprises:

a temporal entity object builder to receive a temporal logic entity from said modeling and testing component,

wherein said temporal entity object builder is adapted to divide said temporal logic entity to at least one component said temporal logic entity is built from; and

wherein said at least one component is sent to a builder within said object builder.

45. A dynamic testing apparatus comprising:

a requirements tests manager to define, simulate, and analyze a scenario;

a simulator manager to coordinate the simulation sequence of said scenario;

a dynamic verification manager to activate said checkers during said simulation sequence; and

a simulation and verification manager to control said dynamic testing component.

46. The dynamic testing apparatus of claim 45, wherein said simulation and verification manager is further adapted to load at least one entity for simulation from said specification repository component.

47. The dynamic testing apparatus of claim 46, wherein said at least one entity can be simulated substantially immediately.

48. The dynamic testing apparatus of claim 45, wherein said simulation and verification manager is further adapted to analyze said scenario.

49. The dynamic testing apparatus of claim 45, wherein said simulation and verification manager is further adapted to initiate said simulator manager and load said checkers for said dynamic verification manager.

50. The dynamic testing apparatus of claim 45, wherein said simulator manager further comprises:
an evaluation component to provide an infrastructure required to execute said scenario; and
an evaluation context component to control an execution of at least one evaluator of said scenario.
51. The dynamic testing apparatus of claim 48, wherein said at least one evaluator is executed in parallel to at least a second evaluator.
52. The dynamic testing apparatus of claim 48, wherein said evaluator evaluates said at least one entity written in said requirements engineering language.
53. The dynamic testing apparatus of claim 48, further comprises an outcome repository to store the outcome of said execution of said at least one evaluator of said scenario.
54. The dynamic testing apparatus of claim 45, wherein said dynamic verification manager is further adapted to detect an error of said execution by said at least one evaluator of said scenario.
55. The dynamic testing apparatus of claim 45, wherein said dynamic verification manager is further adapted to receive values of said at least one entity during simulation.
56. The dynamic testing apparatus of claim 45, wherein said dynamic verification manager is further adapted to check if said values are true values for said simulation.
57. The dynamic testing apparatus of claim 45, wherein said dynamic verification manager is further adapted to detect an error of said execution by said at least one evaluator of said scenario.